**UNIVERSIDAD POLITECNICA SALESIANA**

**NOMBRE: OSCAR MORA**

**GRUPO: 1**

**FECHA: 10-ago.-19**

# MONTECARLO METHOD

The MonteCarlo method is a non-deterministic or numerical statistical method, used to approximate complex and expensive mathematical expressions to evaluate accurately. The method was named in reference to the Monte Carlo Casino (Monaco) for being "the capital of gambling", since roulette is a simple random number generator. The name and systematic development of Monte Carlo's methods date back to approximately 1944 and were greatly improved with the development of the computer.

## CHARACTERISTICS

1. Very simple structure algorithm.

2. The error of the value obtained as a proportional rule

3. Are a class of computational algorithms.

4. Provide generally approximate solutions.

## MONTECARLO DOUBLE INTEGRAL

## Example

$$f(x) = \int_4^5 \int_5^7 \frac{\frac{cos(x)}{2+y}}{log(8+x+y)+cos(x)+sin(y)+cos(sin(log(x+y)))})dxdy$$

## CODE

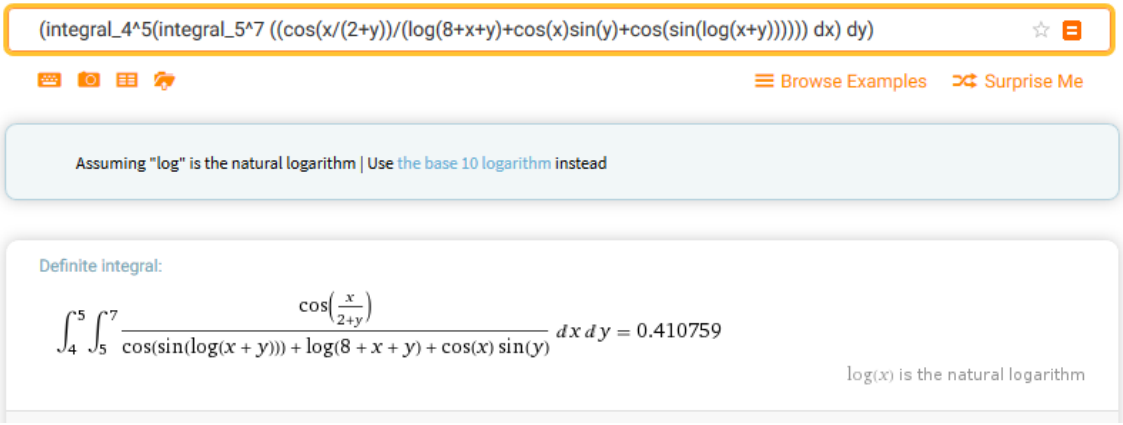
```
from matplotlib.ticker import LinearLocator, FormatStrFormatter
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import matplotlib.pyplot as pp
from matplotlib import cm
import numpy as np
```

```
import random
import math


valor_real=7.79



def FXY(a,b,c,d):
   x=(b-a)*random.random()+a
   y=(d-c)*random.random()+c
   return
((np.cos(x/(2+y)))/((np.log(8+x+y))+(np.multiply(np.cos(x),np.sin(y)))+(np.sin(np.cos(np.log(x
+y))))))



def integral(a, b, c, d):
   print("Mensaje Error")
   suma = 0
   suma1 = 0
   for i in range(N):
      aux = FXY(a,b,c,d)
      suma += aux
      suma1 += math.pow(aux,2)
   calculoIntegral = ((b - a) * (d - c) / N) * suma
   error = (1 / N) * suma1
   error_real =  (b - a) * (d - c) * math.sqrt((calculoIntegral-error)/N)
   return calculoIntegral, error_real


N=100000
total_errores=[]
res,e=integral(4,5,5,7)
print("Error Real",e)
print("Resultudao Integral calculado",res)
```

**RESULT**

```
Mensaje Error
Error Real 0.004885397288734943
Resultudao Integral calculado 0.7315484706611945
```

**WOLFRAMALPHA**




**LINK VIDEO**

https://youtu.be/vkerzu_n6SM

**3D GRAPHIC**

**%matplotlib notebook**

**fig = plt.figure()**

**ax = fig.gca(projection='3d')**

**X = np.arange(-20, 20, 0.25)**

**Y = np.arange(-20, 20, 0.25)**

**X, Y = np.meshgrid(X, Y)**

**Z = ((np.cos(X/(2+Y)))/((np.log(8+X+Y))+(np.multiply(np.cos(X),np.sin(Y)))+(np.sin(np.cos(np.log(X+Y))))))**

**surf = ax.plot_surface(X, Y, Z, cmap=cm.coolwarm,linewidth=0, antialiased=False)**

**ax.set_zlim(0., 115)**

**ax.zaxis.set_major_locator(LinearLocator(10))**

**ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))**

```
for Xi in range(0, 18, 3):
    for Yi in range(-10, 10, 3):
        Zi = ((np.cos(Xi/(2+Yi)))/((np.log(8+Xi+Yi))+(np.multiply(np.cos(Xi),np.sin(Yi)))+(np.sin(np.cos(np.log(Xi+Yi))))))
        if (Zi >= 0):
            for i in range(0, int(Zi), 1):
                ax.scatter(Xi, Yi, i, marker="o", color="yellow",alpha=0.25)


#barra de colores
fig.colorbar(surf, shrink=0.5, aspect=5)
plt.show()
```